**JOBSHEET 4**

**Tugas 1**

package Search;

import java.util.Scanner;

import java.util.Random;

public class BinarySearch {

public static void main(String[] args)

{

Random acak = new Random();

int[] data = new int[10];

Scanner baca = new Scanner(System.in);

int i, step, temp, n = data.length;

int low, mid, high, index, flag, cari;

for (i = 0; i < 10; i++)

{

data[i] = acak.nextInt(201);

}

// BUBBLE SHORT

{

for (step = 1; step < n; step++)

{

int swap = 0;

for(i = 0; i < n-step; i++)

{

if(data[i] > data[i+1])

{

temp = data[i];

data[i] = data [i+1];

data[i+1] = temp;

swap = 1;

}

}

}

for(i = 0; i < data.length; i++)

{

System.out.print(data[i] + " ");

}

System.out.println("");

// BINARY SEARCH

n = data.length;

low = 0;

high = n - 1;

index = 0;

flag = 0;

System.out.print("Masukkan bilangan yang dicari: ");

cari = baca.nextInt();

while(low <= high)

{

mid = (low + high) / 2;

if(cari == data[mid])

{

flag = 1;

index = mid;

break;

}

else if(cari < data[mid])

{

high = mid - 1;

}

else

{

low = mid + 1;

}

}

if(flag == 1)

System.out.println("Data ditemukan di index ke-" + index);

else

{

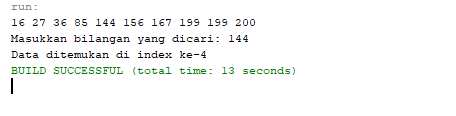
System.out.println(" Data tidak ditemukan " );

}

}

}

}



**Tugas 2**

package linearsearch;

import java.util.Scanner;

import java.util.Random;

public class Linearsearch {

public static void main(String[] args)

{

Scanner baca = new Scanner(System.in);

Random acak = new Random();

int [] data = new int[10];

int i;

int low, mid, high, index, flag, cari, N;

for (i = 0; i < 10; i++)

{

data[i] = acak.nextInt(201);

}

for (i = 0; i < data.length; i++ )

{

System.out.print(data[i] + "" + ",");

}

N = data.length;

low = 0;

high = N-1;

index = 0;

flag = 0;

System.out.println("");

System.out.print("Masukkan bilangan yang di cari: ");

cari = baca.nextInt();

while(low <= high)

{

mid = (low + high / 2);

if(cari == data[mid])

{

flag = 1;

index = mid;

break;

}

else if (cari < data[mid])

{

high = mid-1;

}

else

{

low = mid + 1;

}

}

if(flag == 1)

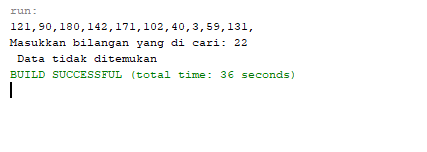
System.out.println("Data ditemukan di index ke-" + index);

else

System.out.println(" Data tidak ditemukan " );

}

}



**JOBSHEET 5**

**Merge Sort**

**Tugas 1**

package mergesort;

public class Tugas01 {

public static void merge(int arr[], int l, int m, int r)

{

int n1 = m - l + 1;

int n2 = r - m;

int L[] = new int[n1];

int R[] = new int[n2];

for(int i = 0; i < n1; i++)

L[i] = arr[l + i];

for(int i = 0; i < n2; i++ )

R[i] = arr[m + 1 + i];

int i = 0, j = 0;

int k = l;

while (i < n1 && j <n2)

{

if (L[i] <= R[j])

{

arr[k] = L[i];

i++;

}

else

{

arr[k] = R[j];

j++;

}

k++;

}

while (i < n1)

{

arr[k] = L[i];

i++;

k++;

}

while (j < n2)

{

arr[k] = R[j];

j++;

k++;

}

}

// METHOD SORT

public static void sort(int arr[], int l, int r)

{

if (l < r)

{

int m = (l+r)/2;

sort(arr, l, m);

sort(arr, m+1, r);

merge(arr, l, m, r);

}

}

//Method printArray

public static void printArray(int arr[])

{

int n = arr.length;

for (int i=0; i<n; ++i)

System.out.print(arr[i] + " ");

System.out.println();

}

public static void main(String args[])

{

int arr[] = new int[] {42, 2, 32, 6, 10, 22, 50, 4};

System.out.println("Sebelum diurutkan");

printArray(arr);

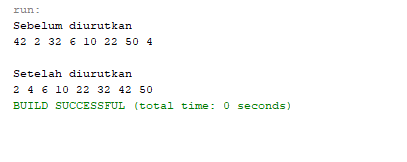
sort(arr, 0, arr.length-1);

System.out.println("\nSetelah diurutkan");

printArray(arr);

}

}



**TUGAS 2**

package mergesort;

import java.util.Scanner;

public class Tugas02 {

public static int[] data;

public static void merge(int arr[], int l, int m, int r)

{

int n1 = m - l + 1;

int n2 = r - m;

int L[] = new int[n1];

int R[] = new int[n2];

for(int i = 0; i < n1; i++)

L[i] = arr[l + i];

for(int i = 0; i < n2; i++ )

R[i] = arr[m + 1 + i];

int i = 0, j = 0;

int k = l;

while (i < n1 && j <n2)

{

if (L[i] <= R[j])

{

arr[k] = L[i];

i++;

}

else

{

arr[k] = R[j];

j++;

}

k++;

}

while (i < n1)

{

arr[k] = L[i];

i++;

k++;

}

while (j < n2)

{

arr[k] = R[j];

j++;

k++;

}

}

// METHOD SORT

public static void sort(int arr[], int l, int r)

{

if (l < r)

{

int m = (l+r)/2;

sort(arr, l, m);

sort(arr, m+1, r);

merge(arr, l, m, r);

}

}

//Method printArray

public static void printArray(int arr[])

{

int n = arr.length;

for (int i=0; i<n; ++i)

System.out.print(arr[i] + " ");

System.out.println();

}

public static void main(String args[]) {

Scanner baca = new Scanner(System.in);

int [] data = new int[10];

{

System.out.println("Algoritma Mergesort");

}

System.out.println("");

for(int i = 0; i < data.length; i++){

System.out.print("Masukkan Bilangan ke-" + " " + (i+1)+ " : " );

data [i] = baca.nextInt();

}

System.out.println("");

System.out.println("Sebelum diurutkan");

printArray(data);

sort(data, 0, data.length-1);

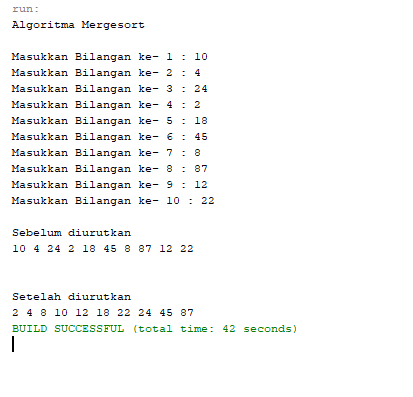
System.out.println("");

System.out.println("\nSetelah diurutkan");

printArray(data);

}

}



**JOBSHEET 6**

**QUICK SORT**

**TUGAS 1**

package QuickSort;

public class Tugas01 {

public static int partition(int arr[], int L, int R)

{

int pivot = arr[R];

int i = (L-1);

for (int j=L; j<R; j++)

{

if (arr[j] <= pivot)

{

i++;

int temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

}

int temp = arr[i+1];

arr[i+1] = arr[R];

arr[R] = temp;

return i+1;

}

// METHOD SORT

public static void sort(int arr[], int L, int R)

{

if (L < R)

{

int pi = partition(arr, L, R);

sort(arr, L, pi-1);

sort(arr, pi+1, R);

}

}

// METHOD printArray

public static void printArray(int arr[])

{

int n = arr.length;

for (int i=0; i<n; ++i)

System.out.print(arr[i]+" ");

System.out.println("");

}

public static void main(String args[])

{

int arr[] = new int[] {10, 2, 30, 26, 45, 4, 6, 20, 22, 15};

int n = arr.length;

System.out.println("Sebelum Diurutkan");

printArray(arr);

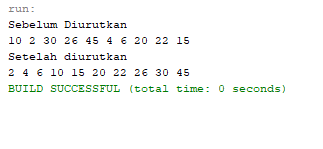
sort(arr, 0, n-1);

System.out.println("Setelah diurutkan");

printArray(arr);

}

}



**Tugas 2**

package QuickSort;

import java.util.Random;

public class Tugas02 {

public static int[] data;

public static int partition(int arr[], int L, int R)

{

int pivot = arr[R];

int i = (L-1);

for (int j=L; j<R; j++)

{

if (arr[j] <= pivot)

{

i++;

int temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

}

int temp = arr[i+1];

arr[i+1] = arr[R];

arr[R] = temp;

return i+1;

}

// METHOD SORT

public static void sort(int arr[], int L, int R)

{

if (L < R)

{

int pi = partition(arr, L, R);

sort(arr, L, pi-1);

sort(arr, pi+1, R);

}

}

// METHOD printArray

public static void printArray(int arr[])

{

int n = arr.length;

for (int i=0; i<n; ++i)

System.out.print(arr[i]+" ");

System.out.println("");

}

public static void main(String args[])

{

data = new int[10];

Random acak = new Random();

int i;

for(i=0; i < 10; i++)

{

data [i] = acak.nextInt(201);

}

System.out.println("Algoritma Quicksort");

System.out.println("");

int n = data.length;

System.out.println("Sebelum Diurutkan");

printArray(data);

System.out.println("");

sort(data, 0, n-1);

System.out.println("Setelah diurutkan");

printArray(data);

}

}

